

**CLAIMS**

1. A method of instrumenting a COM object invoked by a client for performing a  
5 selected business logic, comprising:  
intercepting a request from the client for creating said COM object,  
generating a wrapper object corresponding to said requested COM object, said  
wrapper object implementing a universal interface having a plurality of virtual functions each  
indexed by a number corresponding to an index number of a method associated with an  
10 interface of said requested COM object,  
providing said client with a reference pointer to said wrapper COM object,  
upon invocation of a method associated with an interface of the requested COM  
object by the client, invoking a virtual function of said universal interface of the wrapper  
object indexed by a number corresponding to an index number of said requested method,  
15 wherein said invoked virtual function references instructions for saving a start time  
marker, executing instructions corresponding to said requested method, and saving a stop  
time marker upon completion of execution of said instructions associated with the requested  
method.
- 20 2. The method of claim 1, further comprising registering said invoked method with an  
ARM agent upon invocation of said method by the client.
3. The method of claim 2, wherein saving a start time marker comprises invoking said  
ARM agent for generating a record for a transaction corresponding to invocation of said  
25 method of the requested COM object.
4. The method claim 3, wherein saving a stop time marker comprises invoking the ARM  
agent to generate said stop time marker.
- 30 5. The method of claim 1, wherein said wrapper COM object comprises a data structure  
for storing a number of arguments and a type of each argument associated with each method  
of said requested COM object.

6. The method of claim 1, wherein said wrapper object comprises a reference pointer for referring to said requested COM object.

7. The method of claim 1, wherein said wrapper object comprises a reference pointer for referring to said universal interface.

8. The method of claim 1, further comprising  
defining a policy that indicates whether to perform said step of generating a wrapper  
COM object corresponding to said requested COM object.

9. The method of claim 8, wherein said policy identifies a requested proxy object, a  
COM object belonging to an MTS package, and a COM<sup>+</sup> object for wrapping.

10. The method of claim 8, further comprising  
storing said policy in a tabular format in a registry of a system on which said COM  
objects are executed.

11. The method of claim 1, wherein the step of intercepting a request comprises  
patching code associated with one or more selected system functions.

12. The method of claim 11, wherein said system functions are provided in a dynamic  
link library.

13. The method of claim 12, wherein said dynamic library is Ole32.dll and Mtxex.dll.

14. The method of claim 11, wherein said selected system functions can be any of  
CoInitialize, CoInitializeEx, OleInitialize, CoAddRefServerProcess, CoCopyProxy,  
CoCreateInstance, CoCreateInstanceEx, CoGetCallContext, CoGetClassObject,  
CoGetInstanceFromFile, CoGetInstanceFromFileFromStorage, CoGetInterfaceAndReleaseStream,  
GetObjectContext (NT only), CoGetObjectContext (W2K only), CoImpersonateClient,  
CoInitializeSecurity, CoMarshalInterThreadInterfaceInStream, CoMarshalInterface,  
CoRegisterClassObject, CoReleaseServerProcess, CoRevertToSelf, CoRevokeClassObject,  
CoSetProxyBlanket, CoUninitialize, CoUnmarshalInterface, OleUninitialize.

15. The method of claim 11, further comprising  
for each of said selected system functions, utilizing a hook associated therewith to  
refer to a program for patching said system function.

5

16. The method of claim 15, wherein said hook comprises a designated string stored in  
system registry.

17. The method of claim 16, wherein said patching of the system function comprises  
10 replacing selected bytes in a code corresponding to said system function with a jump  
instruction to a code for creating said requested COM object and generating said wrapper  
object.

18. The method of claim 17, further comprising  
15 copying instructions in said system function code corrupted by said inserted jump  
instruction to an allocated data area.

19. The method of claim 18, further comprising  
decoding said selected bytes prior to their replacement by said jump instruction to  
20 determine a total number of bytes corresponding to instructions corrupted by said inserted  
jump instruction.

20. A system for monitoring a response time of a transaction performed by one or more  
COM objects executing on one or more platforms, comprising  
25 one or more monitoring agents deployed on selected ones of said platforms, each  
monitoring agent intercepting a request for creating at least one of said COM objects and  
generating a wrapper object corresponding to said requested COM object,  
wherein said wrapper object implements a universal interface having a plurality of  
virtual functions each indexed by a number corresponding to an index number of a method  
30 associated with an interface of said requested COM object.

21. The system of claim 20, further comprising  
an ARM agent in communication with said monitoring agent.

22. The system of claim 21, wherein each of said virtual functions refers to a code having instructions for invoking said ARM agent prior to execution of an invoked method of a wrapped COM object to save a start time marker, executing said invoked method, and  
5 invoking said ARM agent subsequent to completion of execution of said invoked method to save a stop time marker.

23. A computer readable medium storing instructions for performing a method of instrumenting a COM object invoked by a client comprising intercepting a request from the  
10 client for creating said COM object, generating a wrapper object corresponding to said requested COM object, said wrapper object implementing a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object, and providing said client with a reference pointer to said wrapper COM object,

15